

CMSC335

Web Application Development with JavaScript



Router, Cookies, Sessions

Department of Computer Science

University of MD, College Park

Slides material developed by Ilchul Yoon, Nelson Padua-Perez

REST (Representational State Transfer)

- **An architectural style**; not a protocol
 - Designed to operate with resource-oriented services (locate/manipulate resource)
 - Allow different data formats (e.g., html, text, JSON)
 - Advantages: Fast, language, and platform independent
- **Resources are represented by URLs**
 - Resource - document, person, location
 - Each resource has a unique URL
 - Each resource can be dynamically generated instead of having an actual page/document
- Operations are performed via HTTP methods (GET, POST, PUT, DELETE) with resources

Express Application Generator

- Link: <https://expressjs.com/en/starter/generator.html>

Router

- To manage the complexity of many routes in your main app, you can define files that take care of some of them. In the file you create a router that will take care of some of those routes
- The Router object (that can handle .get, .post, etc.) is created using `express.Router()`
- Here is an example of a router for requests that start with **/building**

`/* Code in file building.js */`

`const express = require('express');`

`let router = express.Router();`

`router.get("/", (request, response) => { response.send("/ in building.js") });`

`router.get("/iribe", (request, response) => { response.send("/iribe in building.js") });`

- For the above code in the main app we have

`app.use("/buildings", buildings);`

- **Example:** Router

Cookies

- Cookie - small piece of information sent by a server and stored either in the browser's memory or as a small file in the hard drive. Acceptance of the cookie depends on the client
- **Browser sends the cookie back with every request to the server that sent the cookie**
- Cookie - contains a name/value pair. This is how the cookie information may look like when sent by the server in the http header

`Set-Cookie: automobile=nelyota; path=/; domain=notRealCars.comr`

- Setting a cookie - associating a value with a name
- Getting a cookie - getting the value associated with a name
- Cookie size - 4KB per cookie

Cookies with an Expiration Date

- Cookies without an expiration date will expire when the browser is closed
- Specify expiration date using “expires” and date in GMT
- GMT (Greenwich mean time)
 - Wdy, DD-Mon-YYYY HH:MM:SS GMT
 - Sun, 15-Apr-2016 11:29:00 GMT
- **Example:** `setCookieExpiration.html`
 - Syntax is very strict (you must have space after semicolon)
- To delete a cookie set the expiration time to some point in the past
- **Example:** Cookies (in Node)

Sessions

- **Session** - time period during which a person views a number of different web pages in a browser and then quits
- **What would you like**
 - To keep track of information throughout the session. For example, keeping track of color preferences, usernames, data selection, etc.
- **What is the problem?**
 - http (the protocol that makes possible the communication between browsers and web servers) is stateless
 - Stateless - every page request is independent
- **Solution** - The concept of sessions
- **Example: Sessions**
 - Let's try with one or two users
 - Let's see the cookie with session information